

CHAPTER 1

INTRODUCTION

1.1 Introduction

A 3D interactive graphics application is an extremely computational demanding paradigm, requiring the simulation and display of a virtual environment at interactive frame rates. It is significant in real time game environment. Even with the use of powerful graphics workstations, a moderately complex virtual environment can involve a vast amount of computation, inducing a noticeable lag into the system. This lag can detrimentally affect the visual effect and may therefore severely compromise the diffusion of the quality of graphics application.

Therefore, a lot of techniques have been proposed to overcome the delay of the display. It includes motion prediction, fixed update rate, visibility culling, frameless rendering, Galilean antialiasing, level of detail, world subdivision or even employing parallelism. Researches have been done and recovered that the fixed update rates and level of detail technique are the only solutions which enable the application program the balance the load of the system in real-time (Reddy, 1997). Of these solutions, concentration is focused on the notion of level of detail.

Since the mid nineteen-seventies, programmers have used Level of Detail (LOD) management to improve the performance and quality of their graphics systems. The LOD approach involves maintaining a set of representations of each polygonal object, each with varying levels of triangle resolution. During the

execution of the animation, object deemed to be less important is displayed with a low-resolution representation. Whereas object of higher importance is displayed with higher level of triangle resolution.

The drastic growth in scanning technology and high realism computer simulation complexity has led to the increase of dataset size. Only super computer or powerful graphics workstation are capable to handle these massive datasets. For this reason, problem of dealing with meshes that are apparently larger than the available main memory exists. Data, which has hundreds, million of polygons are impossible to fit in any available main memory in desktop personal computer. Because of this memory shortage, conventional simplification methods, which typically require reading and storing the entire model in main memory, cannot be used anymore. Hence, out-of-core approaches are gaining its attention widely.

As commonly known, graphics applications always desire high realism scene yet smooth scene rendering. Smooth rendering can be achieved by reducing the number of polygons to a suitable level of detail using the simplification technique. It saves milliseconds of execution time that help to improve performance. However, in order to obtain a nice simplified mesh, surface attributes other than geometry information are essential to be preserved as well. Eye catching surface appearance certainly will increase the beauty of the scene effectively.

1.2 Background Research

Traditionally, polygonal models have been used in computer graphics extensively. Till this moment, large variety of applications is using this fundamental primitive to represent three dimensional objects. Besides, many graphics hardware and software rendering systems support this data structure. In addition, all virtual environment systems employ polygon renderers as their graphics engine.

In reality, many computational demanding systems desire smooth rendering of these polygonal meshes. To optimize the speed and quality of graphics rendering,

level of details has been used widely to reduce the complexity of the polygonal mesh using level of detail technique. In short, a process which takes an original polygon description of a three dimensional object and creates another such description, retaining the general shape and appearance of the original model, but containing fewer polygons.

Recent advances in scanning technology, simulation complexity and storage capacity have lead to an explosion in the availability and complexity of polygonal models, which often consist of millions of polygons. Because of the memory shortage in dealing with meshes that are significantly larger than available main memory, conventional methods, which typically require reading and storing the entire model in main memory during simplification process, cannot solve the dilemma anymore. Thus, out-of-core approaches are introduced consequently.

Out-of-core algorithms are also known as external algorithms or secondary memory algorithms. Out-of-core algorithms keep the bulk of the data on disk, and keep in main memory (or so called in-core) only the part of the data that's being processed. Lindstrom (Lindstrom, 2000a) is the pioneer in out-of-core simplification field. He created a simplification method; called OoCS which is independent of input mesh size. However, the output size of the mesh must be smaller than the available main memory. Later on, other researchers carried out similar approaches.

Especially in out-of-core simplification, a large number of research have been done on level of detail's construction and management for use in interactive graphics applications, mostly in medical visualization, flight simulators, terrain visualization systems, computer aided design and computer games. For instance, simplification is used broadly in medical and scientific visualization. It always involves a lot of processing on high resolution three dimensional data sets. The data is mainly produced by those high technology scanners, such as CT or MRI scanners. The simplification process may need to extract volumetric data at different density levels. If the accuracy is not that critical, one may only process on its isosurfaces. Anyhow, these data simplification require a lot of processing time and it is mainly run daily on supercomputers worldwide.

Graphics applications, which demand high accuracy in simplification development is critical. It is essential to maintain the high quality and good frame rates at the same time. For example, medical visualization and terrain visualization is crucial in maintaining a good visual fidelity. Anyway, in many real time systems, the quality of data visualization has to be degraded in order to retain superior rendering time. For instance, an excellent frame rate is vital in game environment without doubt. Thus, the quality of simplified model has to be sacrificed sometimes.

Rendering the large models at interactive frame rates is essential in many areas, includes entertainment, training, simulation and urban planning. Out-of-core techniques are required to display large models at interactive frame rates using low memory machines. Hence, it needs new solution or further improvement such as prefetching, geomorphing, appearance preservation, parallelization, visibility pre-computing, geometry caching, image-based rendering, and etc.

To avoid the last minute data fetching when needed, prefetching, visibility pre-computing and geometry caching are imperative. Although the changes from frame to frame are regularly small, however, they are occasionally large, so, prefetching technique is needed. This technique predicts or speculates which part of the model are likely to become visible in the next few frames then prefetch them from disk ahead of time. Correa *et al.* (2002) showed that prefetching can be based on from-point visibility algorithms. Visibility can be pre-computed using from-region visibility or from-point visibility. Whilst geometry caching exploits the coherence between frames, thus keeping geometry cache in main memory and update the cache as the viewing parameters changes.

Above and beyond, parallelization and image-based rendering enhance the frame rates as well. Parallelization (Correa *et al.*, 2002) uses a few processors to run different tasks at the same time. Or, it can make use of multithreading concept in a single-processor machine too. On the other hand, image-based rendering techniques (Wilson and Monacha, 2003) such as texture-mapped impostors can be used to accelerate the rendering process. These texture-mapped impostors are generated either in a preprocessing step or at runtime (but not every frame). These techniques are suitable for outdoor models.

The geomorphing and surface preserving are potential in pleasant scene rendering. An unfortunate side effect of rendering with dynamic levels of detail is the sudden visual ‘pop’ that occurs when triangles are inserted or removed from the mesh. Geomorphing allows smooth transitions between the approximations (Levenberg, 20002; Erikson, 2000; Hoppe, 1998a). In virtual environments or three dimensional game engines, the surface attributes play an important role to make the object looks attractive. Therefore, these surface attributes like colors and textures should be maintained after simplification process.

1.3 Motivations

Why do we care about visualization of large datasets? Due to the advances in scanning technology and complexity of computer simulation, the size of the datasets grows rapidly these years. The data is vital because it has application in many areas, such as computer design and engineering, visualization of medical data, modeling and simulation of weapons, exploration of oil and gas, virtual training and many more.

These massive data can only be rendered on high end computer system. If it is needed to run on personal computer, it may be an impossible mission, or even it can, the output is jagged or ungraceful. Therefore, to run it on expensive high end graphics machine, it is very cost ineffective and not user friendly. There is a need to display the data in low cost PC with high quality output.

Surface attributes, for example, normal, curvature, color and texture values are important to make the rendered objects looks attractive. It can increase the realism of a virtual environment. It shows the details of an object, such as its illumination, lighting effect and material attributes. Without it, the rendered scene will become dull and bored.

1.4 Problem Statement

The datasets are getting enormous in size. However, even the well implemented in-core methods no more able to simplify these massive datasets. This is mainly because in-core approach loads the whole full resolution mesh into main memory during simplification process. Besides, we cannot keep relying on high end graphics machine as it is expensive and not everyone has the chance to use it. Therefore, when the datasets bigger than the main memory, the datasets cannot be simplified.

Geometry aspects like vertex position always retained after simplification process whether in in-core simplification or out-of-core simplification. However, work in preserving the surface appearance, e.g. surface normal, curvature and color or even texture attributes in the original mesh is not common in out-of-core simplification approach. The lost surface attributes will greatly reduce the realism of virtual environment.

1.5 Purpose

To render the massive datasets in 3D real-time environment and preserve its surface appearance during simplification process using commodity personal computer.

1.6 Objectives

1. To develop an out-of-core simplification technique.
2. To preserve the surface attributes on the out-of-core model based on error metrics.

1.7 Research Scope

- a) Only triangular polygonal mesh is considered, other data representation is not investigated here.
- b) The simplification is for only static polygonal objects, dynamic object is not covered here.
- c) Only vertex positions, normals, colors and texture coordinates are preserved after simplification process.
- d) Application is run on commodity personal computer. Commodity in this content means low cost PC with not more than 2GB RAM and not any kind of SGI machine.
- e) Graphics card of the PC is assumed capable in handling real time rendering.
- f) Only simple level of detail management system is applied by using distance criterion.
- g) Secondary memory used here is the hard disk, other secondary memory devices are not investigated its cons and pros.
- h) The size of the datasets mustn't larger than the size of secondary memory owned by the machine.
- i) No other enhancement techniques such as geomorphing, prefetching, caching, parallelization and no disk usage reduction are investigated.